Python bling bling

Victor Stinner

Pycon FR, Paris, mai 2009

- 1 < x <= 10
- 0 <= x < y <= 20
- a, b = b, a + b # Fibonacci
- color = "red" if load > 0.8 else "green"

```
>>> carres = [1, 4, 9, 16, 25]
>>> carres[1:-1]
[4, 9, 16]
>>> carres[::2]
[1, 9, 25]
>>> carres[::-1]
[25, 16, 9, 4, 1]
```

- str, unicode, list, tuple, dict, set, <fichier>, ...
- dict.iterkeys(), dict.itervalues(), dict.iteritems()
- iter(object) ou "for element in object: ..."
- Votre objet : méthode __iter__()

```
>>> list(enumerate("abc"))
[(0, 'a'), (1, 'b'), (2, 'c')]
>>> list(zip("ab", "XY"))
[('a', 'X'), ('b', 'Y')]
>>> gen = itertools.count()
>>> gen.next(), gen.next()
(0, 1)
```

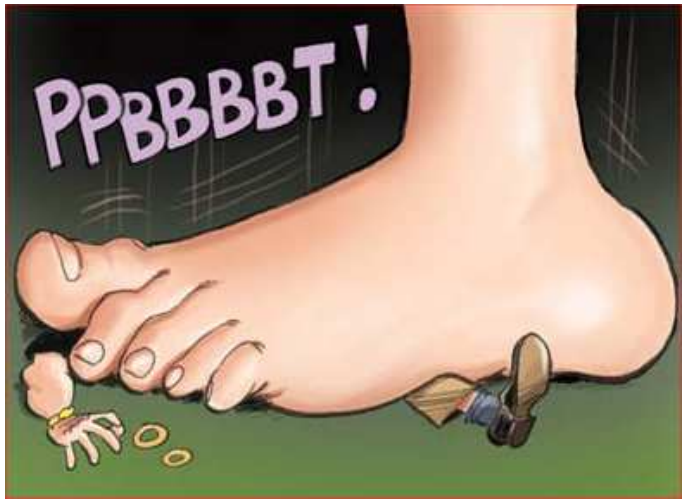
- (item[2] for item in items)
- imap(operator.itemgetter(2), items)
- (item.name for item in items)
- imap(operator.attrgetter("name"), items)

```
def fibonacci():
  u, v = 1, 0
  while 1:
    yield u
    u, v = v, u + v
```

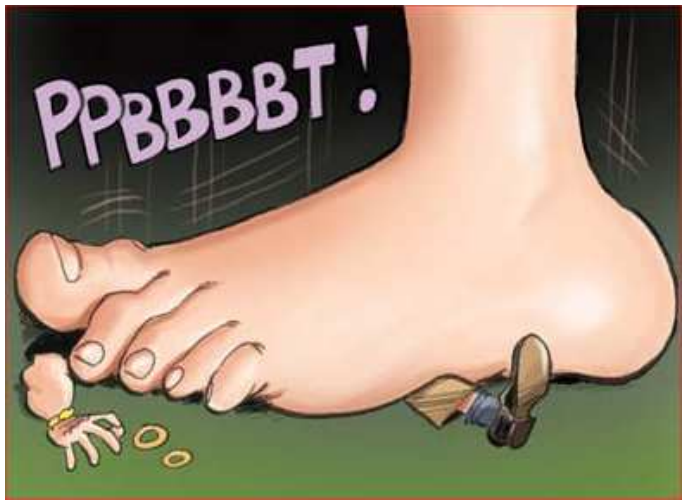
- Simple
- Rapide
- Générique
- Suite infinie

- list : [(3 + x**2) for x in xrange(10, 1000)]
- tuple : tuple((3 + x**2) for x in xrange(10, 1000))
- dict : dict((cle, valeur) for cle, valeur in elements)
- ...

```
f = open('text', 'w')
f.write('hello!')
f.close()
```



```
# Python 2.5
from __future__ \
 import with_statement
with open('text', 'w') as f:
    f.write('hello!')
```



Fichier, verrou, vos propres objets, etc.

- @classmethod
- @staticmethod
- @property

```
def deprecated(comment=None):
    def _deprecated(func):
        def newFunc(*args, **kwargs):
            message = "Call deprecated function %s"
            if comment:
                message += ": " + comment
            warn(message, category=DeprecationWarni
            return func(*args, **kwargs)
        newFunc.__name__ = func.__name__
        newFunc.__doc__ = func.__doc__
        return newFunc
    return _deprecated
```

```
@deprecated
def initialize_random():
    ...

@deprecated("Use TimedeltaWin64 field type")
def durationWin64(field):
    ...
```

```
>>> "a" in set("abc")
True
>>> set("abc") | set("ad")
set(['a', 'c', 'b', 'd'])
>>> set("abc") - set("ad")
set(['c', 'b'])
```

- datetime : date et heure
- logging : loguer du texte
- optparse : arguments sur la ligne de commande
- pprint.pprint() : *pretty print*

- BeautilfulSoup : parser du HTML non conforme
- PIL : Python Imaging Library

- http://upload.wikimedia.org/wikipedia/en/9/90/Chapman_as_Brian
- http://www.flickr.com/photos/chris-parry/3007001331/
- http://www.flickr.com/photos/barrykidd/1763691970/
- http://stackoverflow.com/questions/101268
- http://python.net/ goodger/projects/pycon/2007/idiomatic/
- Trucs et astuces, Victor Stinner, Hors Série Linux Magazine 40 (janvier/février 2009)